

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: TIME-BASED NETWORK CONNECTIONS

APPLICANT: DAYLAN B. DARBY

CERTIFICATE OF MAILING BY EXPRESS MAIL

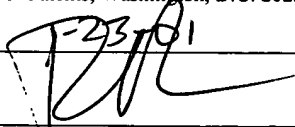
Express Mail Label No. EL724384600US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

Date of Deposit

Signature

Typed or Printed Name of Person Signing Certificate

F-23701

Rich Donovan

TIME-BASED NETWORK CONNECTIONS

BACKGROUND

This invention relates to time-based network connections.

A communication protocol generally defines a format for providing end-to-end service (e.g., data delivery) between two end-points connected to a network (e.g., a public network such as the Internet or a private network such as a local intranet), usually independent of the characteristics of the particular network. Some protocols, such as transmission control protocol (TCP) and user datagram protocol (UDP), typically do not complete transactions between the two end-points until a network connection between the two end-points is properly terminated. If the network connection is lost or otherwise ends improperly, then during the next network connection between the two end-points, the transactions may be repeated.

DESCRIPTION OF DRAWINGS

FIG. 1 shows a simplified network configuration.

FIG. 2 is a flowchart showing a process of connecting to a server.

DESCRIPTION

Referring to FIG. 1, a network configuration 100 includes a workstation 102 that can communicate with a server 104 across a network 106 using communication links 108 and 110.

5 When a client 112 included in the workstation 102 establishes a connection with the server 104, a process 114 included in the client 112 sets a timer, tracking the duration of the connection. After the client 112 completes a transaction with the server 104, the process 114 determines if a certain amount of time has elapsed as indicated by the timer. If the certain amount of time has not elapsed, the client 112 and the server 104 remain connected and can continue communicating. If the amount of time has elapsed, the process 114 triggers the termination of the connection between the client 112 and the server 104. The client 112 can then immediately attempt to establish another connection with the server 104.

10 Terminating the connection between the client 112 and the server 104 after a certain amount of time reduces the risk of unexpected connection loss between the client 112 and the server 104 as compared to a connection of unlimited duration or to a connection limited by other criteria such as number of allowable transactions per connection session. (Other criteria may be used in addition to the time-based criteria.) Terminating the connection also enables the server 104 to

perform any procedures it normally performs upon termination of a connection, such as committing (completing) the transaction with the client 112, thereby reducing the chances of the server 104 later duplicating the transaction with the client 112 should the connection unexpectedly terminate.

These procedures on the server 104 are typically governed by the protocol(s) implemented by the server 104, e.g., transport protocols such as TCP, UDP, and Internet Protocol (IP), electronic mail (email) protocols, and other similar types of protocols.

For example, referring to FIG. 2, the process 114 begins 200 when the workstation 102 connects 202 to the server 104. The workstation 102 that includes the client 112 can include any device capable of running the client 112 and connecting to the network 106, such as a mobile computer, a stationary computer, a server, a personal digital assistant, a telephone, a pager, or other similar device. The network 106 can include any kind and any combination of networks such as the Internet, a local network, a private network, a public network, or other similar network. The client 112 includes one or more programs that enable a user of the workstation 102 to interact with the workstation 102. The server 104 can include any device capable of connecting to the network 106 and storing data. The communication links 108 and 110 between the workstation

102 and the server 104 can be any kind and any combination of communication links such as modem links, cables, point-to-point links, infrared connections, fiber optic links, cellular links, Bluetooth, satellite links, or other similar links. The workstation 102 and the server 104 may be remotely located from each other, capable of communicating only across the network 106 using the communication links 108 and 110.

(The network configuration 100 is simplified for ease of explanation; the network configuration 100 may include additional elements such as additional networks, proxy servers, firewalls or other security mechanisms, Internet Service Providers (ISPs), and other elements.)

The process 114 can be included as part of the client 112 as shown, be otherwise included on the workstation 102 (e.g., as a stand-alone application or as part of another application), or be otherwise accessible to the workstation 102 (e.g., be included on a network accessible by the workstation 102 and/or the client 112 or be part of a network stack that the workstation 102 uses to communicate with the network 106 such as a TCP/IP protocol stack).

The client 112 could include an electronic mail (email) application that enables the user to send and receive email messages across the network 106 using the server 104 as a "host," the location that hosts (stores) the user's incoming

email messages at least until the user retrieves the email messages. Examples of email applications in a UNIX context include elm and pine. The client 112 can run in any operating system environment, such as UNIX, Linux, Windows, and other similar operating systems. The server 104 and/or the client 112 may be capable of implementing any version of Post Office Protocol (POP), Internet Message Access Protocol (IMAP), Application Configuration Access Protocol (ACAP), Simple Mail Transfer Protocol (SMTP), extended SMTP (ESMTP), or other similar protocol. POP, IMAP, ACAP, SMTP, and ESMTP are protocols that the server 104 and/or the client 112 may use to process, send, and/or receive email messages. Examples of transactions that the server 104 may not complete until the connection is properly terminated include deleting files, designating files for deletion, moving files, sending files, and other similar operations.

Once the workstation 102 connects to the server 104, typically through the client 112, the process 114 sets 204 a timer. The workstation 102 or the client 112 may notify the process 114 that a connection has been established with the server 104, or the process 114 may be configured to detect when a connection has been established (e.g., by running as a daemon, continually monitoring for establishment of a connection). The timer may be set to zero (or other value)

and count up, or it may be set with a predetermined value and count down. The predetermined value reflects an amount of time that the workstation 102 and the server 104 may remain connected before their connection should be terminated. The timer's set value may be based on ordinary time (hours, minutes, seconds, etc.) or on another counting system (clock ticks, etc.). In any event, once set, the timer begins counting up or down as appropriate to track the elapsed time of the present connection between the workstation 102 and the server 104.

Using the email application as the example transaction here, the client 112 asks 206 the server 104 (possibly through the process 114) whether the server 104 has an email message available for the user. If the server 104 has no available email messages, then the workstation 102 disconnects 208 from the server 104. The server 104 may not have any available messages because the user has no new email messages, the server 104 does not presently have access to stored email messages, or other similar reason. When the connection between the workstation 102 and the server 104 is terminated, the process 114 may clear 210 the timer. In that case, the process 114 may reset the timer and need not set the timer as described above when (or if) the workstation 102 establishes a

new connection with the server 104; the process 114 need only start the timer counting up or down as appropriate.

The connection between the workstation 102 and the server 104 may automatically terminate if no email messages are available, or the user may be given the option to remain connected. If the user chooses to remain connected, then the timer may keep running or it may be cleared and reset. Once the workstation 102 disconnects from the server 104, the process 114 ends 212.

If the server 104 does have an available email message for the user, then the client 112 retrieves 214 the email message from the server 104. The client 112 retrieves the email message using any technique supported by the client 112 and the workstation 102, such as by downloading the email message across the network 106 over the communication links 108 and 110. After retrieving the email message, the client 112 requests 216 that the server 104 delete the email message. Alternatively, depending on the type of client and/or on the server's protocol, the client 112 may not need to separately request the deletion or to request the deletion at all.

The process 114 then determines 218 if the timer has expired. If the timer was counting down from a predetermined value, then the timer has expired if the timer value equals (or is less than) zero or other stop value. If the timer was

counting up, then the timer has expired if the timer equals
(or exceeds) a stop value. The stop value, like the
predetermined value described above, reflects an amount of
time that the workstation 102 and the server 104 may remain
5 connected before their connection should be terminated.

If the timer has not expired, then the client 112 asks
206 the server 104 whether the server 104 has an email message
available for the user as described above.

If the timer has expired, then the workstation 102
10 disconnects 220 from the server 104. As described above, when
the connection between the workstation 102 and the server 104
is terminated, the process 114 may clear 222 the timer and
reset the timer, thus meaning that the process 114 need not
set the timer when (or if) the workstation 102 establishes a
new connection with the server 104. Once terminated from the
server 104, the workstation 102 may then automatically or
manually connect 202 again to the server 104.

The reestablishment of a connection between the
workstation 102 and the server 104 may be performed
20 transparently to the user, thus reducing disruption to the
user by not requiring repeated logging in to the client 112
and/or the server 104. The termination and subsequent
establishment of a connection with the server 104 can be
performed with little or no visual disruption or processing

interruption to the user. On the other hand, the process 114 may terminate the connection and provide notice to the user, such as by displaying a message box on the workstation's display screen indicating that the connection to the server 104 has been terminated. The user may be required to manually trigger a new connection to the server 104, such as by clicking on or selecting a button in the message box.

Once the workstation 102 is no longer connected to the server 104, depending on the protocol used, the process 114 may trigger the deletion of any mail retrieval code(s) used on the workstation-side in retrieving email messages from the server 104. The deletion of the code(s) can help maintain security on the workstation-side, preventing another user at the workstation 102 from using a previous user's mail retrieval code(s) to illicitly retrieve the previous user's email messages from the server 104.

Once the server 104 is no longer connected to the workstation 102, the server 104 can perform its usual post-connection procedures (if any). These post-connection procedures may include deleting the retrieved email message(s), perhaps in response to the client's request(s) to delete the retrieved email message(s). Thus, the server 104 can delete the email message(s) retrieved by the workstation 102 during the previous connection session, thereby reducing

or eliminating the chances that the server 104 would
mistakenly interpret the retrieved email messages as
unretrieved (as may happen with improper or unexpected
termination of the connection) and that the workstation 102
5 would download email messages it already retrieved from the
server 104 (upon reestablishing connection with the server
104).

The techniques described here are not limited to any
particular hardware or software configuration; they may find
10 applicability in any computing or processing environment. The
techniques may be implemented in hardware, software, or a
combination of the two. The techniques may be implemented in
programs executing on programmable machines such as mobile or
stationary computers, personal digital assistants, telephones,
15 pagers, and similar devices that each include a processor, a
storage medium readable by the processor (including volatile
and non-volatile memory and/or storage elements), at least one
input device, and one or more output devices. Program code is
20 functions described and to generate output information. The
output information is applied to one or more output devices.

Each program may be implemented in a high level
procedural or object oriented programming language such as C
to communicate with a machine system such as a computer

system. However, the programs can be implemented in assembly or machine language, if desired. In any case, the language may be a compiled or interpreted language.

Each such program may be stored on a storage medium or device, e.g., compact disc read only memory (CD-ROM), hard disk, magnetic diskette, or similar medium or device, that is readable by a general or special purpose programmable machine for configuring and operating the machine when the storage medium or device is read by the machine to perform the procedures described in this document. The system may also be considered to be implemented as a machine-readable storage medium, configured with a program, where the storage medium so configured causes a machine to operate in a specific and predefined manner.

Other embodiments are within the scope of the following claims.